UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/163,041	09/29/1998	DANIEL P. VEDITZ	013.0067	9527

26171 7590 05/04/2007

FISH & RICHARDSON P.C.
P.O. BOX 1022
MINNEAPOLIS, MN 55440-1022

EXAMINER	
FRINK, JOHN MOORE	
ART UNIT	**PAPER NUMBER**
2142	

MAIL DATE	DELIVERY MODE
05/04/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	09/163,041	VEDITZ, DANIEL P.	
	Examiner	Art Unit	
	John M. Frink	2142	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *23 October 2006*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *33-81* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *33-81* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

Claim Rejections - 35 USC § 102

1. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless – (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

Claim Rejections - 35 USC § 103

2. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

3. Claims 33, 34, 36 – 39, 41 - 42, 44, 46, 47, 49 – 52, 54 - 55, 57, 58, 60 and 61 – 64, 66 - 70, 72 - 74 are rejected under 35 U.S.C. 102(b) as anticipated by Englander (Developing Java Beans) or, in the alternative, under 35 U.S.C. 103(a) as obvious over Englander in view of java.sun.com (http://java.sun.com/docs/tutorial/jar/basics/manifest.html; Applications Bundled as JAR Files section).

4. Regarding claim 33, Englander shows a method for executing an application that is encapsulated in a package, the method comprising:

loading the package (represented as the directory holding the referenced HTML file and the referenced .jar file; Section 6.3.1) within a browser on a local client computer (Section 1.2.2), the package including a manifest (represented by the <APPLET> tag

and its contents; Section 6.3.1) and an archive of files (represented by the BeansBook.jar file; Section 6.3.1) that include instructions and content needed to execute the application (represented by the CODE section of the <APPLET> tag Section 6.3.1); the archive of files including: an initial file that includes instructions for initiating execution of the application, and other files needed to execute the application (represented by the file containing the <APPLET> tag; Section 6.3.1), and the manifest including an initial file identifier that indicates that the initial file is to be processed before the other files in the archive of files when the application is executed (represented by the <APPLET> tag, specifically the CODE section that points to PickleUser.class; Section 6.3.1);

in response to loading the package within the browser, automatically accessing the manifest; locating the initial file identifier in the manifest; based on the located initial file identifier, accessing the instructions for initiating the execution of the application in the initial file; processing the accessed instructions; and automatically initiating execution of the application based on the processed instructions (represented by automatically loading and executing PickleUser.class; Section 6.3.1);.

Alternatively, Englander shows a method for executing an application that is encapsulated in a package, the method comprising:

loading the package within a browser on a local client computer, the package including a manifest and an archive of files that include instructions and content needed to execute the application, the archive of files including:

an initial file that includes instructions for initiating execution of the application (Sections 6.1 and 6.2), and other files needed to execute the application (Section 6.1, represented by 'files . . .' in the jar command, Paragraph 1 – 2),

Englander does not show the manifest including an initial file identifier that indicates that the initial file is to be processed before the other files in the archive of files when the application is executed; in response to loading the package within the browser, automatically accessing the manifest; locating the initial file identifier in the manifest; based on the located initial file identifier, accessing the instructions for initiating the execution of the application in the initial file; processing the accessed instructions; and automatically initiating execution of the application based on the processed instructions.

Java.sun.com shows the manifest including an initial file identifier that indicates that the initial file is to be processed before the other files in the archive of files when the application is executed in response to loading the package within the browser, automatically accessing the manifest; locating the initial file identifier in the manifest; based on the located initial file identifier, accessing the instructions for initiating the execution of the application in the initial file; processing the accessed instructions; and automatically initiating execution of the application based on the processed instructions (Applications Bundled as JAR Files Section; specifically represented by use of the Main-Class header).

It would have been obvious to one of ordinary skill in the art at the time of the invention to modify the disclosure of Englander with that of java.sun.com as both disclosure's elaborate on Java's JAR file format.

5. Regarding claim 46, Englander shows a computer program product for executing an application that is encapsulated in a package, the computer program product being embodied in a computer readable medium and including instructions that, when executed by a processor, cause the processor to:

load the package within the computer program product (represented as the directory holding the referenced HTML file and the referenced .jar file; Section 6.3.1), the package including a manifest (represented by the <APPLET> tag and its contents; Section 6.3.1) and an archive of files (represented by the BeansBook.jar file; Section 6.3.1) that include instructions and content needed to execute the application (represented by the file containing the <APPLET> tag; Section 6.3.1), the archive of files including: an initial file that includes instructions for initiating execution of the application, and other files needed to execute the application (represented by the file containing the <APPLET> tag; Section 6.3.1), and the manifest including an initial file identifier that indicates that the initial file is to be processed before the other files in the archive of files when the application is executed; automatically access the manifest in response to loading the package within the computer program product (represented by the <APPLET> tag, specifically the CODE section that points to PickleUser.class; Section 6.3.1); locate the initial file identifier in the manifest;

access the instructions for initiating the execution of the application in the initial file

based on the located initial file identifier; process the accessed instructions; and automatically initiate execution of the application based on the processed instructions (represented by automatically loading and executing PickleUser.class; Section 6.3.1).

Alternatively, Englander shows a computer program product for executing an application that is encapsulated in a package, the computer program product being embodied in a computer readable medium and including instructions that, when executed by a processor, cause the processor to: load the package within the computer program product, the package including a manifest and an archive of files that include instructions and content needed to execute the application, the archive of files including: an initial file that includes instructions for initiating execution of the application, (Sections 6.1 and 6.2), and other files needed to execute the application (Section 6.1, represented by 'files . . . ' in the jar command, Paragraph 1 – 2).

Englader does not show the manifest including an initial file identifier that indicates that the initial file is to be processed before the other files in the archive of files when the application is executed; automatically access the manifest in response to loading the package within the computer program product; locate the initial file identifier in the manifest; access the instructions for initiating the execution of the application in the initial file based on the located initial file identifier; process the accessed instructions; and automatically initiate execution of the application based on the processed instructions.

Java.sun.com shows show the manifest including an initial file identifier that

indicates that the initial file is to be processed before the other files in the archive of files when the application is executed; automatically access the manifest in response to loading the package within the computer program product; locate the initial file identifier in the manifest; access the instructions for initiating the execution of the application in the initial file based on the located initial file identifier; process the accessed instructions; and automatically initiate execution of the application based on the processed instructions (Applications Bundled as JAR Files Section; specifically represented by use of the Main-Class header).

It would have been obvious to one of ordinary skill in the art at the time of the invention to modify the disclosure of Englander with that of java.sun.com as both disclosure's elaborate on Java's JAR file format.

6. Regarding claims 34 and 47, Englander further shows the method for executing an application that is encapsulated in a package of claim 33, further comprising receiving the package at the local client computer (6.3.1, where BeansBook.jar is downloaded).

7. Regarding claims 36 and 49, Englander further shows accessing the additional instructions in the one or more other files; and processing the additional instructions, the accessing and processing of the additional instructions being performed in response to processing the instructions for initiating execution of the application in the initial file (6.1, 6.2 and 6.3.1, where multiple .jar files can be used, and where each .jar has, by default, a Manifest file specifying additional instructions).

8. Regarding claims 37 and 50, Englander further shows where the initial file comprises a source file for a web page, the instructions for initiating execution of the application in the initial file comprise instructions for rendering the web page, and automatically initiating execution of the application comprises rendering the web page in accordance with the instructions for rendering the web page (6.3 and 6.3.1, where the HTML files shown comprise a standard web page source file, which is inherently rendered when it is run).

9. Regarding claims 38 and 51, Englander further shows where the source file for the web page comprises an HTML document (6.3 and 6.3.1).

10. Regarding claims 39 and 52, Englander further shows where the initial file comprises an executable file, and the instructions for initiating execution of the application in the initial file comprise program execution instructions (6.3.1, where PickleUser.class, contained in BeansBook.jar, is referenced. Futhermore, PickleUser.class contains program instructions executable by web browsers and appletviewer).

11. Regarding claims 41 and 54, Englander further shows where the instructions for initiating execution of the application in the initial file comprise JAVA-based instructions (6.3.1, where BeansBook.jar contains the Java-based instructions in PickleUser.class).

12. Regarding claims 42 and 55, Englander further shows where the manifest further comprises an archive type identifier that identifies an application type of the application (6.3.1, where identifies the ARCHIVE as of type '.jar', which further contains a file of identified as being of the type '.class'), and processing the accessed instructions

comprises processing the accessed instructions in accordance with the application type of the application (6.3.1, where appletviewer runs files in accordance with types .html, and .jar).

13. Regarding claims 44 and 58, Englander further shows where the initial file identifier indicates that the initial file is the first file to be processed in the archive of files when the application is executed (6.3.1, paragraphs 3 – 4).

14. Regarding claim 57, Englander further shows where the computer program product comprises a browser (Section 1.2.2).

15. Regarding claims 45 and 59, Englander further shows where automatically initiating execution of the application based on the processed instructions comprises automatically initiating execution of the application without maintaining a connection between the local client computer and a web server (6.3.1 where the .jars are downloaded, 6.4 where downloaded .jars are executed without requiring said connection as all required items are on users local "C:" drive).

16. Regarding claim 60, Englander shows method for encapsulating an application in a package such that the application may be automatically executed by a browser (Section 1.2.2) on a client computer, the method comprising:

generating an archive of files (representing by generating the .jar archive, Section 6.1, further referenced in 6.3.1, which is a part of the folder that contains the HTML file containing the <APPLET> tag referenced in 6.3.1) that include instructions and content needed to execute the application (represented by the <APPLET> tag in 6.3.1, specifically the CODE element), the archive including:

an initial file that includes instructions for initiating execution of the application (represented by the HTML file in 6.3.1, specifically the CODE element of the <APPLET> tag), and other files needed to execute the application (represented by the content of the .jar file, 6.1, 6.2 and 6.3.1) and generating a manifest file that is associated with the archive, the manifest file including an initial file identifier that instructs the browser to process the initial file before processing other files in the archive in order to initiate execution of the application (represented by the CODE element of the <APPLET> tag, which specifies executing the PickleUser.class file before any others in order to initiate execution of the BeansBook.jar application).

Alternatively, Englander shows generating an archive of files that include instructions and content needed to execute the application (represented by the .jar file of Sections 6.1, 6.2 and 6.3.1, and its included Manifest file, included by default, Section 6.2), the archive including other files needed to execute the application (6.1, 6.2), and generating a manifest file that is associated with the archive (6.2).

Englander does not show the archive including: an initial file that includes instructions for initiating execution of the application, and the manifest file including an initial file identifier that instructs the browser to process the initial file before processing other files in the archive in order to initiate execution of the application.

Java.sun.com shows the archive including: an initial file that includes instructions for initiating execution of the application, and the manifest file including an initial file identifier that instructs the browser to process the initial file before processing

other files in the archive in order to initiate execution of the application (Applications Bundled as JAR Files Section; specifically represented by use of the Main-Class header).

It would have been obvious to one of ordinary skill in the art at the time of the invention to modify the disclosure of Englander with that of java.sun.com as both disclosure's elaborate on Java's JAR file format.

17. Regarding claim 61, Englander further shows where the instructions for initiating execution of the application in the initial file include instructions to process additional instructions from one or more of the other files in the archive (6.1, 6.2 and 6.3.1, where multiple .jar files can be used, and where each .jar has, by default, a Manifest file specifying additional instructions).

18. Regarding claim 62, Englander further shows where the initial file comprises a source file for a web page, and the instructions for initiating execution of the application in the initial file comprise instructions for rendering the web page (6.3 and 6.3.1, where the HTML files shown comprise a standard web page source file, which is inherently rendered when it is run).

19. Regarding claim 63, Englander further shows where the source file for the web page comprises an HTML document (6.3 and 6.3.1).

20. Regarding claim 64, Englander further shows where the initial file comprises an executable file, and the instructions for initiating execution of the application in the initial file comprise program execution instructions (6.3.1, where PickleUser.class, contained

in BeansBook.jar, is referenced. Futhermore, PickleUser.class contains program instructions executable by web browsers and appletviewer).

21. Regarding claim 66, Englander further shows where the instructions for initiating execution of the application in the initial file comprise JAVA-based instructions (6.3.1, where BeansBook.jar contains the Java-based instructions in PickleUser.class).

22. Regarding claim 67, Englander further shows where the manifest file further comprises an archive type identifier that instructs the browser to process the instructions for initiating execution of the application in the initial file in accordance with the application type of the application (6.3.1, where identifies the ARCHIVE as of type '.jar', which further contains a file of identified as being of the type '.class'; and 6.3.1, where appletviewer runs files in accordance with types .html, and .jar).

23. Regarding claim 68, Englander shows a self-contained package for distributing an application, the package being embodied in a computer readable medium (represented as the directory holding the referenced HTML file and the referenced .jar file; Section 6.3.1), and configured to enable the application to be automatically executed within a browser environment (1.2.2), the package comprising:

an archive (represented by the BeansBook.jar file, 6.3.1) including:

an initial content source having instructions for initiating execution of the application within the browser environment (represented by the CODE section of the <APPLET> tag Section 6.3.1), and

additional files that include instructions and content needed to execute the application within the browser environment (represented by the file containing the <APPLET> tag;

Section 6.3.1); and a manifest associated with the archive that includes an initial content identifier that indicates that the initial content source is to be processed before the additional files in the archive when the application is executed (represented by the <APPLET> tag, specifically the CODE section that points to PickleUser.class; Section 6.3.1).

Alternatively, Englander shows a self-contained package for distributing an application (represented by the .jar file, 6.1, 6.2, 6.3.1), the package being embodied in a computer readable medium and configured to enable the application to be automatically executed within a browser environment (1.2.2.), the package comprising: an archive including: an initial content source having instructions for initiating execution of the application within the browser environment, and additional files that include instructions and content needed to execute the application within the browser environment; and a manifest associated with the archive (represented by the jar files Manifest file, which is included with a .jar file by default; 6.1 – 6.2).

Englander does not show where the .jar file includes an initial content identifier that indicates that the initial content source is to be processed before the additional files in the archive when the application is executed.

Java.sun.com shows where the .jar file includes an initial content identifier that indicates that the initial content source is to be processed before the additional files in the archive when the application is executed (Applications Bundled as JAR Files Section; specifically represented by use of the Main-Class header).

It would have been obvious to one of ordinary skill in the art at the time of the invention to modify the disclosure of Englander with that of java.sun.com as both disclosure's elaborate on Java's JAR file format.

24. Regarding claim 69, Englander further shows where said initial content source is an HTML file containing content layout instructions for rendering a document that displays content included in the archive (6.3 and 6.3.1, where said layout instructions are represented by the WIDTH and HEIGHT elements of the <APPLET> tag).

25. Regarding claim 70, Englander further shows where said initial content source is an executable file containing program execution instructions (6.3.1, where the executable instructions are contained within the <APPLET> tag).

26. Regarding claim 72, Englander further shows where the manifest further comprises an archive type identifier that identifies an application type of the application (6.3.1, where identifies the ARCHIVE as of type '.jar', which further contains a file of identified as being of the type '.class').

27. Regarding claim 73, Englander further shows where the archive is configured according to a .JAR type file structure (6.2, 6.3.1, specifically BeansBook.jar).

28. Regarding claim 74, Englander further shows where the additional files comprise one or more of a web page, a script, an image, a sound file, and a JAVA file (6.2.1).

29. Claims 35 and 48 are rejected under 35 U.S.C. 103(a) as being unpatentable over Englander in view of Rowe et al. (5,964,836).

Englander shows the method of claim 34 and the computer program product of claim 47.

Englander does not show where receiving the package at a local client computer comprises loading the package onto the local client computer from a local computer readable medium.

Rowe et al. shows where receiving the package at a local client computer comprises loading the package onto the local client computer from a local computer readable medium (col. 6 lines 47 – 50).

It would have been obvious to one of ordinary skill in the art at the time of the invention to modify the disclosure of Englander with that of Rowe et al. in order to utilize a standard and convenient method of loading an application on a computer.

30. Claims 40, 53, 65 and 71 are rejected under 35 U.S.C. 103(a) as being unpatentable over Englander in view of Bates et al. (5,877,766).

Englander shows the methods of claim 39 and 65, the computer program product of claim 52, and the self-contained package of claim 70.

Englander does not show where the executable file is a JAVAscript file.

Bates et al. shows where an executable file is a JAVAscript file (col. 6 lines 37 – 59).

It would have been obvious to one of ordinary skill in the art at the time of the invention to modify the disclosure of Englander with that of Bates et al. in order to utilize a standard and common executable file format.

31. Claims 43 and 56 are rejected under 35 U.S.C. 103(a) as being unpatentable over Englander in view of Oran et al. (5,617,526).

Englander shows the methods of claim 33 and the computer program product of claim 45.

Engalnder does not show displaying an icon associated with the application, wherein loading the package in the browser or other computer program product comprises automatically loading the package in the browser or computer program product in response to a user selecting the icon associated with the application.

Oran et al. shows display an icon associated with an application, wherein loading the package in the browser or other computer program product comprises automatically loading the package in the browser or computer program product in response to a user selecting the icon associated with the application (col. 4 lines 19 – 38).

It would have been obvious to one of ordinary skill in the art at the time of the invention to modify the disclosure of Englander with that of Oran et al. in order to utilize a standard method of identifying and launching a browser or other computer program product.

32. Claims 75 – 77 and 79 are rejected under 35 U.S.C. 103(a) as being unpatentable over Englander in view of Renshaw (6,065,024).

33. Regarding claim 75, Englader shows an application executing an application encapsulated in a self-contained package comprising (6.3.1): an input module for inputting the self-contained package, wherein the self-contained package includes (6.2): an archive including (6.2, 6.3.1, said archive represented by the .jar file)
an initial content source having instructions for initiating execution of the application, and additional files that include instructions and content needed to execute the

application (6.2, 6.3.1, represented by the HTML file containing the <APPLET> tag, and, additionally by said .jar files Manifest), and a manifest associated with the archive that includes an initial content identifier that indicates that the initial content source is to be processed before the additional files in the archive when the application is executed (6.1, 6.2); a processing engine for accessing the manifest, locating the initial file identifier in the manifest (6.1 – 6.3.1, specifically where the Java runtime environment executes the jar file, or, alternatively, when embedded within HTML, the Java runtime environment is called through the use of 'appletviewer'), based on the located initial file identifier, accessing the instructions for initiating execution of the application in the initial file, and processing the instructions for initiating execution of the application in the initial file before processing the instructions in the additional files (6.1 – 6.3).

Englander does not show where said application for executing said encapsulated application is a web browser, or a content rendering and layout module for rendering content according to the instructions in the initial content source and the additional files.

Renshaw shows where said application for executing said encapsulated application is a web browser, or a content rendering and layout module for rendering content according to the instructions in the initial content source and the additional files (col. 2 lines 9 – 62).

It would have been obvious to one of ordinary skill in the art at the time of the invention to modify the disclosure of Englander with that of Renshaw in order to utilize a web browsers, applications common at the time of the invention, for executing said Java .jar packages.

34. Regarding claim 76, Englander in view of Renshaw further show where said initial content source is an HTML file containing content layout instructions for rendering a document that displays content included in the archive (Englander, 6.3.1, where said instructions are the WIDTH and HEIGHT elements within the <APPLET> tag).

35. Regarding claim 77, Englander in view of Renshaw further show where said initial content source is an executable file containing program execution instructions (6.3.1, where PickleUser.class, contained in BeansBook.jar, is referenced. Futhermore, PickleUser.class contains program instructions executable by web browsers and appletviewer).

36. Regarding claim 79, Englander in view of Renshaw further show said executable file includes JAVA-based instructions (6.3.1).

37. Claim 78 is rejected under 35 U.S.C. 103(a) as being unpatentable over Englander in view of Renshaw as applied to claim 77 above, and further in view of Bates et al.

Englander in view of Renshaw show the web browser according to claim 77.

Englander in view of Renshaw do not show where said executable file includes Javascript instructions.

Bates et al. show a web browser executing Javascript (col. 6 lines 37 – 59).

It would have been obvious to one of ordinary skill in the art at the time of the invention to further modify the disclosure of Englander with that of Bates et al. in order to support another computer programming language, commonly utilized in web browsers at the time of the invention.

38. Claims 80 and 81 are rejected under 35 U.S.C. 103(a) as being unpatentable over Englander in view of Renshaw as applied to claim 75 above, and further in view of Rowe et al.

39. Regarding claim 80, Englander in view of Renshaw show the web browser according to claim 75.

Englander in view of Renshaw do not show where the input module is configured to input the self-contained software package from a computer readable medium.

Rowe et al. shows where the input module is configured to input the self-contained software package from a computer readable medium (col. 6 lines 47 – 50).

It would have been obvious to one of ordinary skill in the art at the time of the invention to further modify the disclosure of Englander with that of Rowe et al. in order to utilize a method that was both standard and convenient at the time of the invention.

40. Regarding claim 81, Englander in view of Renshaw, further in view of Rowe et al. show where said computer readable medium is a compact disc containing digitally recorded information (Rowe et al., col. 6 lines 47 – 50).

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to John M. Frink whose telephone number is (571) 272-9686. The examiner can normally be reached on M-F 7:30AM - 5:00PM EST; off alternate Fridays.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Andrew Caldwell can be reached on (571)272-3868. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

John Frink

(571) 272-9686

Andrew Caldwell

ANDREW CALDWELL
SUPERVISORY PATENT EXAMINER